

## EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L4	2	(US-20040133457-\$ or US-20060136904-\$).did.	US-PGPUB	OR	OFF	2007/09/21 19:27
L3	2	L1 not L2	US-PGPUB; USPAT	OR	OFF	2007/09/21 19:26
L2	19	(US-20030069737-\$ or US-20040002838-\$ or US-20060106626-\$ or US-20020165701-\$ or US-20040030786-\$).did. or (US-6002854-\$ or US-5802508-\$ or US-5873081-\$ or US-5515524-\$ or US-6216109-\$ or US-6300948-\$ or US-6405308-\$ or US-5825651-\$ or US-6178502-\$ or US-6241775-\$ or US-6009406-\$ or US-5996114-\$ or US-5615341-\$ or US-6807576-\$). did.	US-PGPUB; USPAT	OR	OFF	2007/09/21 19:26
L1	21	(US-20030069737-\$ or US-20040002838-\$ or US-20060106626-\$ or US-20020165701-\$ or US-20040030786-\$ or US-20040133457-\$ or US-20060136904-\$).did. or (US-6002854-\$ or US-5802508-\$ or US-5873081-\$ or US-5515524-\$ or US-6216109-\$ or US-6300948-\$ or US-6405308-\$ or US-5825651-\$ or US-6178502-\$ or US-6241775-\$ or US-6009406-\$ or US-5996114-\$ or US-5615341-\$ or US-6807576-\$). did.	US-PGPUB; USPAT	OR	OFF	2007/09/21 19:26
S64	2	DAG same (conflict\$4 with (merg\$4))	US-PGPUB; USPAT	OR	ON	2007/09/21 16:19
S61	26	DAG with conflict\$4	US-PGPUB; USPAT	OR	ON	2007/09/21 16:19
S63	0	DAG adj edit	US-PGPUB; USPAT	OR	OFF	2007/09/21 16:15
S62	0	DAG adj edit	USPAT	OR	OFF	2007/09/21 16:15
S60	0	DAG with (conflict\$4 incompatible disjoint inconsistent clash\$ disagree\$6 discord\$4 discrepant incongruous inharmonious) with (remov\$4 prun\$5 chopp\$4 cut\$5 edit\$4)	US-PGPUB; USPAT	OR	ON	2007/09/21 14:31

## EAST Search History

S59	1	DAG with (conflict\$4 incompatible disjoint inconsistent clash\$ disagree\$6 discord\$4 discrepant incongruous inharmonious) same (remov\$4 prun\$5 chopp\$4 cut\$5 edit\$4)	US-PGPUB; USPAT	OR	ON	2007/09/21 11:18
S57	16	S56 and (remov\$4 prun\$5 chopp\$4 cut\$5 edit\$4)	US-PGPUB; USPAT	OR	OFF	2007/09/21 11:15
S58	6	S56 and (remov\$4 prun\$5 chopp\$4 cut\$5 edit\$4) and conflict\$4	US-PGPUB; USPAT	OR	ON	2007/09/21 09:35
S56	19	(US-20030069737-\$ or US-20040002838-\$ or US-20060106626-\$ or US-20020165701-\$ or US-20040030786-\$).did. or (US-6002854-\$ or US-5802508-\$ or US-5873081-\$ or US-5515524-\$ or US-6216109-\$ or US-6300948-\$ or US-6405308-\$ or US-5825651-\$ or US-6178502-\$ or US-6241775-\$ or US-6009406-\$ or US-5996114-\$ or US-5615341-\$ or US-6807576-\$). did.	US-PGPUB; USPAT	OR	OFF	2007/09/21 09:32
S55	1746	(configuration) with (conflict\$4 incompatible disjoint inconsistent clash\$ disagree\$6 discord\$4 discrepant incongruous inharmonious)	USPAT	OR	OFF	2007/09/15 18:09



Search History

Search the Web

Your web history is limited

Web History for [redacted]

All History

Sep 21, 2007

<a href="#">Web</a>	7:23pm	Searched for <a href="#">An Object Model for Evolutionary Configuration Management</a>
<a href="#">Images</a>	7:21pm	Searched for <a href="#">Toward SCM / PDM Integration?</a> -  Viewed 1 result
<a href="#">News</a>		<a href="#">Toward SCM / PDM integration?</a> - imag.fr
<a href="#">Products</a>		
<a href="#">Sponsored Links</a>	4:15pm	Searched for <a href="#">merge DAG Edit</a>
<a href="#">Video</a>	4:14pm	Searched for <a href="#">DAG Edit</a> -  Viewed 2 results
<a href="#">Maps</a>		<a href="#">DAG-Edit Meeting: Save the Date! (fwd)</a> - plantontology.org
<a href="#">Books</a>		<a href="#">http://www.yeastgenome.org/fungi/fungal_anatomy_ontology/faq...</a>
<a href="#">Pause</a>	4:13pm	Searched for <a href="#">DAG Edit</a> -  Viewed 2 results
<a href="#">Remove items</a>		<a href="#">http://genetics.stanford.edu/go-email/email-gofriends/gofrien...</a>
		<a href="#">Software</a> - yeastgenome.org -  See 1 more page
<a href="#">Trends</a>	4:12pm	Searched for <a href="#">DAG Edit</a> -  Viewed 3 results
<a href="#">Interesting Items</a>		<a href="#">SourceForge.net: Files</a> - sourceforge.net
<a href="#">Bookmarks</a>		<a href="#">DAG-Edit Documentation</a> - geneontology.org -  See 1 more page
		<a href="#">http://amigo.geneontology.org/dev/java/dagedit/docs/downloadi...</a>
	4:12pm	Searched for <a href="#">merge DAG Edit</a>
	4:10pm	Searched for <a href="#">merge DAG</a> -  Viewed 2 results
		<a href="#">Optimally Work-Competitive Scheduling for Cooperative...</a> - uconn.edu
		<a href="#">go-2004: Re: Does DAG-Edit understand merging correctly?</a> - stanford.edu -  See 1 more page
	4:09pm	Searched for <a href="#">repair DAG</a>
	9:27am	Searched for <a href="#">combining DAG</a>
	9:26am	Searched for <a href="#">combining DAG</a> -  Viewed 1 result
		<a href="#">The Combining DAG</a> - acm.org
	9:26am	Searched for <a href="#">cobining DAG</a>

[redacted]  
[redacted] [redacted]  
[redacted] [redacted]  
[redacted] [redacted]  
[redacted] [redacted]


Bookmark pages more easily. Install the [free Google Toolbar](#).

[Older](#) [Oldest](#) »

Search History

Search the Web

©2007 Google - [Google Home](#) - [Web History Help](#) - [Privacy Policy](#) - [About Google](#)



# PORTAL

USPTO

**THE ACM DIGITAL LIBRARY**

Subscribe (Full Service)   Register (Limited Service, Free)   [Login](#)

Search:   ☒ The ACM Digital Library   ☐ The Guide

DAG merge

☒ Save results to a Binder  
☒ Search Tips  
☐ Open results in a new window

Sort results by relevance  
 Display expanded form  
 Results 1 - 20 of 200

Best 200 shown

Result page:  
 1   2   3   4   5   6   7   8   9   10   next

Relevance scale:  
 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Terms used: DAG merge

Feedback Report a problem   Satisfaction survey

Found 13,333 of 211,032






☒ **C**   ☐ **R**



Rakesh Ghiya, Laurie J. Hendren  
 January 1996   **Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '96**



Publisher: ACM Press

Full text available: [pdf \(1.51 MB\)](#)

Additional Information: full citation, references, citations, index terms

- 2  **Compilation: Efficient partitioning of fragment shaders for multiple-output hardware**  
Tim Foley, Mike Houston, Pat Hanrahan  
April 2004 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware HHWS '04**  
Publisher: ACM Press  
Full text available:  [pdf\(1483.53 KB\)](#) Additional Information: full citation, abstract, references, citations, index, terms
- 3  **An execution model for limited ambiguity rules and its application to derived data update**  
I.-Min A. Chen, Richard Hull, Dennis McLeod  
December 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 4  
Publisher: ACM Press  
Full text available:  [pdf\(3.36 MB\)](#) Additional Information: full citation, abstract, references, citations, index, terms, review
- 4  **A novel execution model for rule application in active databases is developed and applied to the problem of updating derived data in a database represented using a semantic, object-based database model. The execution model is based on the use of "limited ambiguity rules" (LARs), which permit disjunction in rule actions. The execution model essentially performs a breadth-first exploration of alternative extensions of a user-requested update. Given an object-based database schema, ...**  
**Keywords:** active database systems, deltas on database states, derived data, limited ambiguity rules, semantic data models, update propagation

4  **Sorting on a parallel pointer machine with applications to set expression evaluation**  
Michael T. Goodrich, S. Rao Kosaraju  
March 1996 **Journal of the ACM (JACM)**, Volume 43 Issue 2  
**Publisher:** ACM Press  
Full text available:  pdf(3.04 MB)  
Additional Information: [Full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)  
We present optimal algorithms for sorting on parallel CREW and CREW versions of the pointer machine model. Intuitively, one can view our methods as being based on a parallel mergesort using linked lists rather than arrays (the usual parallel data structure). We also show how to exploit the "locality" of our approach to solve the set expression evaluation problem, a problem with applications to database querying and logic-programming in  $O(\log n) \dots$   
**Keywords:** PRAM, cascade merging, expression evaluation, linking automaton, mergesort, parallel algorithms, pointer machine

5  **Database concurrency control using data flow graphs**  
M. H. Eich, David L. Wells  
June 1988 **ACM Transactions on Database Systems (TODS)**, Volume 13 Issue 2  
**Publisher:** ACM Press  
Full text available:  pdf(2.42 MB)  
Additional Information: [Full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

- used for scheduling database operations, particularly in an OLAP database environment. A DBFG explicitly maintains intertransaction and intratransaction dependencies, and is constructed from the Transaction Flow Graphs (TFG) of active transactions. A TFG, in turn, is the generalization of a query tree used, for example, in DIRECT [15]. All DBFG schedules ...
- 6 Off-line and on-line algorithms for deducing equalities**  
Peter Downey, Hanan Samet, Ravi Sethi  
January 1978 **Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages POPL '78**  
Publisher: ACM Press  
Full text available: [pdf\(1.22 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)
- The classical common subexpression problem in program optimization is the detection of identical subexpressions. Suppose we have some extra information and are given pairs of expressions  $e1=e12$  which must have the same value, and expressions  $f1=f12$  which must have different values. We ask if as a result,  $h1=h2$ , or  $h1 \neq h2$ . This has been called the uniform word problem for finitely presented algebras, an ...
- 7 Session 4 (brief announcements): Optimally work-competitive scheduling for cooperative computing with merging groups**  
Chryssis Georgiou, Alexander Russell, Alex A. Shvartsman  
July 2002 **Proceedings of the twenty-first annual symposium on Principles of distributed computing PODC '02**  
Publisher: ACM Press  
Full text available: [pdf\(125.90 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)
- 8 Optimizing combinatorial library construction via split synthesis**  
Barry Cohen, Steven Skiena  
April 1999 **Proceedings of the third annual international conference on**

# Computational molecular biology RECOMB '99

Publisher: ACM Press  
Full text available: [pdf\(1.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

## Pipelining with futures

Guy E. Blelloch, Margaret Reid-Miller  
June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures SPAA '97**  
Publisher: ACM Press  
Full text available: [pdf\(1.73 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

## Pipelining in multi-query optimization

Nilesh N. Dalvi, Sumit K. Sanghal, Prasan Roy, S. Sudarshan  
May 2001 **Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems PODS '01**  
Publisher: ACM Press  
Full text available: [pdf\(282.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Database systems frequently have to execute a set of related queries, which share several common subexpressions. Multi-query optimization exploits this, by finding evaluation plans that share common results. Current approaches to multi-query optimization assume that common subexpressions are materialized. Significant performance benefits can be had if common subexpressions are pipelined to their uses, without being materialized. However, plans with pipelining may not always be realizable with ...

## Combinational logic synthesis for LUT based field programmable gate arrays

Jason Cong, Yuzheng Ding  
April 1996 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 1 Issue 2  
Publisher: ACM Press  
Full text available: [pdf\(628.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The increasing popularity of the field programmable gate-array (FPGA) technology has generated a great deal of interest in the algorithmic study and tool development for FPGA-specific design automation problems. The most widely used FGAs are LUT based FGAs, in which the basic logic element is a K-input one-output lookup-table (LUT) that can implement any Boolean function of up to K variables. This unique feature of the LUT has brought new challenges to its ...

**Keywords:** FPGA, area minimization, computer-aided design of VLSI, decomposition, delay minimization, delay modeling, logic optimization, power minimization, programmable logic, routing, simplification, synthesis, system design, technology mapping

## A Complete Axiomatization of Full Join Dependencies

Edward Sciore  
April 1982 **Journal of the ACM (JACM)**, Volume 29 Issue 2  
Publisher: ACM Press  
Full text available: [pdf\(1.08 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

## Effectiveness of cross-platform optimizations for a java just-in-time compiler

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=305525&CFTOKEN=41811933>

9/21/2007

Kazuaki Ishizaki, Mikio Takeuchi, Kiyokuni Kawachiya, Toshio Suganuma, Osamu Gohda, Tatsushi Inagaki, Akira Koseki, Kazunori Ogata, Motohiro Kawahito, Toshiaki Yasue, Takeshi Ogasawara, Tammy Onodera, Hideaki Komatsu, Toshio Nakatani  
October 2003 **ACM SIGPLAN Notices, Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11  
Publisher: ACM Press

Full text available: [pdf\(405.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the system overview of our Java Just-In-Time (JIT) compiler, which is the basis for the latest production version of IBM Java JIT compiler that supports a diversity of processor architectures including both 32-bit and 64-bit modes, CISC, RISC, and VLIW architectures. In particular, we focus on the design and evaluation of the cross-platform optimizations that are common across different architectures. We studied the effectiveness of each optimization by selectively disabling ...

**Keywords:** Java, just-in-time compiler, optimization

## Multicore architectures and algorithms: Scheduling threads for constructive cache sharing on CMPs

Shimin Chen, Phillip B. Gibbons, Michael Kozuch, Vasileios Liaskovitis, Anastassia Ailamaki, Guy E. Blelloch, Babak Falsafi, Limor Fix, Nikos Hardavellias, Todd C. Mowry, Chris Wilkerson  
June 2007 **Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures SPAA '07**  
Publisher: ACM Press

Full text available: [pdf\(301.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In chip multiprocessors (CMPs), limiting the number of offchip cache misses is crucial for good performance. Many multithreaded programs provide opportunities for constructive cache sharing, in which concurrently scheduled threads share a largely overlapping working set. In this paper, we compare the performance of two state-of-the-art schedulers proposed for fine-grained multithreaded programs: Parallel Depth First (PDF), which is specifically designed for constructive cache sharing, ...

**Keywords:** chip multiprocessors, constructive cache sharing, parallel depth first, scheduling algorithms, thread granularity, work stealing, working set profiling

## Shading and shaders: Efficient partitioning of fragment shaders for multipass rendering on programmable graphics hardware

Eric Chan, Ren Ng, Pradeep Sen, Kekoa Proudfoot, Pat Hanrahan  
September 2002 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware HWWS '02**  
Publisher: Eurographics Association

Full text available: [pdf\(337.34 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Real-time programmable graphics hardware has resource constraints that prevent complex shaders from rendering in a single pass. One way to virtualize these resources is to partition shading computations into multiple passes, each of which satisfies the given constraints. Many such partitions exist for a shader, but it is important to find one that renders efficiently. We present Recursive Dominator Split (RDS), a polynomial-time algorithm that uses a cost model to find near-optimal partitions of ...

**Keywords:** graph partitioning algorithms, multipass rendering, programmable graphics hardware, shading languages

## Parallel algorithms for evaluating sequences of set-manipulation operations

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=305525&CFTOKEN=41811933>

9/21/2007

 Mikhal J. Atallah, Michael T. Goodrich, S. Rao Kosaraju  
November 1994 **Journal of the ACM (JACM)**, Volume 41 Issue 6


Publisher: ACM Press

Full text available:  pdf(3.00 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Given an off-line sequence  $S$  of  $n$  set-manipulation operations, we investigate the parallel complexity of evaluating  $S$  (i.e., finding the response to every operation in  $S$  and returning the resulting set). We show that the problem of evaluating  $S$  is in NC for various combinations of common set-manipulation operations. Once we establish membership in NC (or, if membership in  $< \dots$

**Keywords:** divide-and-conquer, off-line evaluation, parallel computation, parallel data structures

## 17 Efficient multiple and predicated dispatching

 Craig Chambers, Weimin Chen


October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available:  pdf(2.41 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The speed of message dispatching is an important issue in the overall performance of object-oriented programs. We have developed an algorithm for constructing efficient dispatch functions that combines novel algorithms for efficient single dispatching, multiple dispatching, and predicate dispatching. Our algorithm first reduces methods written in the general predicate dispatching model (which generalizes single dispatching, multiple dispatching, predicate classes and classifiers, and patter ...

## 18 Incremental compilation of optimized code

 Lori L. Pollock, Mary Lou Soffa


January 1985 **Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages POPL '85**

Publisher: ACM Press

Full text available:  pdf(1.57 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Although optimizing compilers have successfully been used to reduce the size and running times of compiled programs, present incremental compilers only support the incremental update of unoptimized code. In this work, we extend the notion of incremental compilation to include optimized code. Techniques to incrementally compile locally optimized code, given intermediate code modifications are developed using a program representation based on flow graphs and dags. A model is designed to repre ...

## 19 Characterization and elimination of redundancy in recursive programs

 Norman H. Cohen


January 1979 **Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages POPL '79**

Publisher: ACM Press

Full text available:  pdf(1.41 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many well-known functions are computed by interpretations of the recursion schemaprocedure  $f(x)$ : if  $p(x)$  then return  $a(x)$  else return  $b(x, f(c_1(x)), \dots, f(c_n(x)))$ . Some of these interpretations define redundant computations because they lead to multiple calls on  $f$  with identical argument values. The existence and nature of the redundancy depend on properties of the functions  $c_i$ . We explore four sets of assumptions about these functions. We analyze directed acyclic ...

## 20 Eliminating Redundant Recursive Calls.


 Norman H. Cohen

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=305525&CFTOKEN=41811933>

9/21/2007

 July 1983 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 5 Issue 3





Publisher: ACM Press

Full text available:  pdf(1.74 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media Player  Real Player

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=305525&CFTOKEN=41811933>

9/21/2007